A recent open source embedded implementation of the DESFire specification designed for on-the-fly logging with NFC based systems

Maxie Dion Schmidt

maxieds@gmail.com
http://people.math.gatech.edu/~mschmidt34
https://github.com/maxieds

Sandia National Labs
Technical Presentation
Spring 2022

High-level overview – NFC

- NFC protocol over short-distance RFID @ 13.56MHz (high-frequency)
- Contactless exchanges between passive tags (PICC) and active hosts (PCD)
- Common applications include:
  - Physical authentication (door readers)
  - University ID cards
  - Bus passes (e.g., ATL Metro MARTA)
  - Renting bikes or motorized scooters
  - Vending machines and other virtual payment kiosks
- Most common tag types exchange data over ISO protocols with wrapped instruction sets

High-level overview – DESFire tags and Chameleon Mini

- DESFire tags: Modern cryptographic algorithms and sophisticated featureset (filesystem, etc.)
- Chameleon Mini (RevG) devices in NFC applications
- DESFire emulation support a popular feature request for the Chameleon Mini

High-level overview – Project big picture and takeaways

- Significance
- Limitations
- Key challenges in development

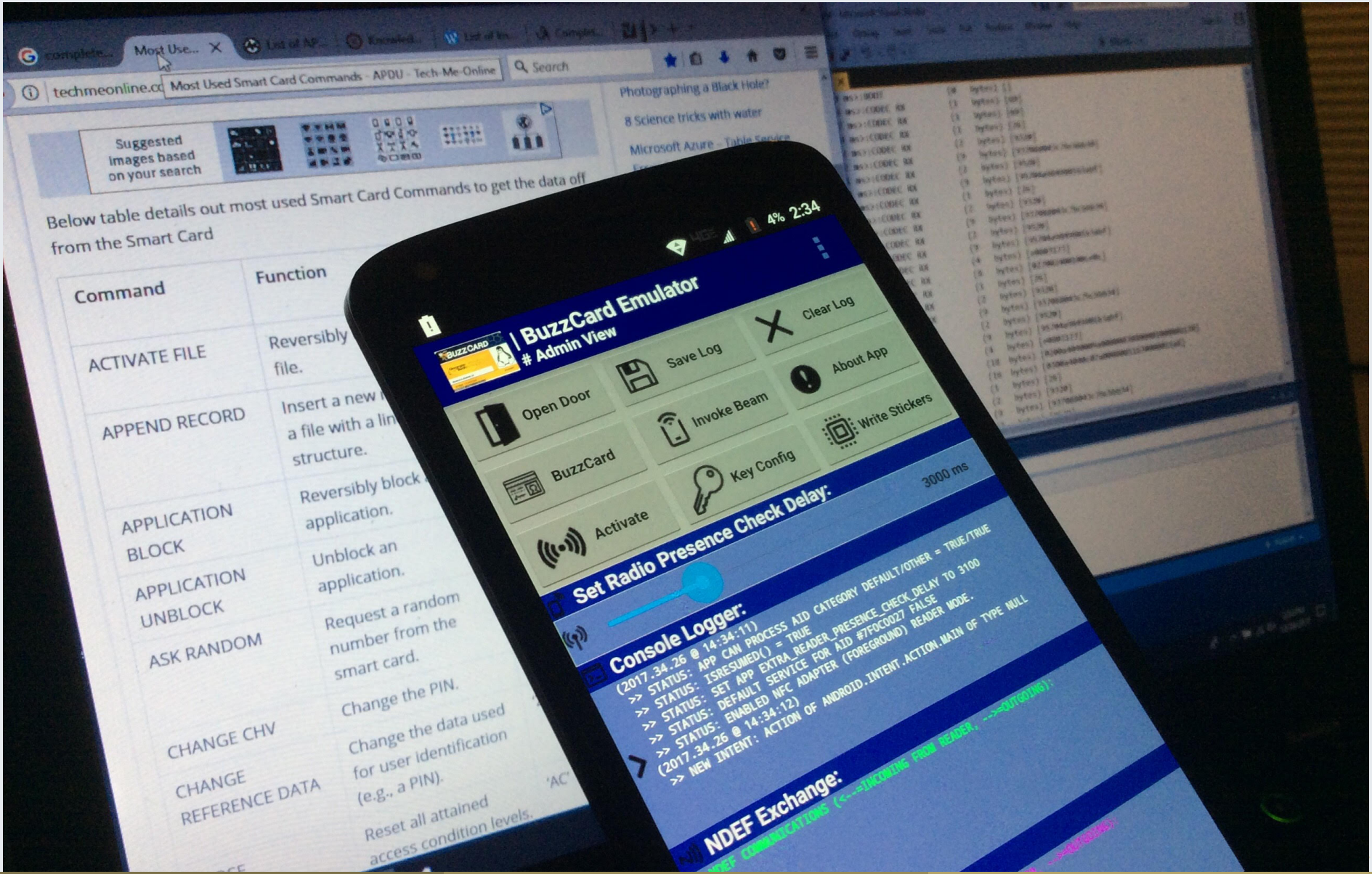
Outline of topics

Presentation outline – Topics

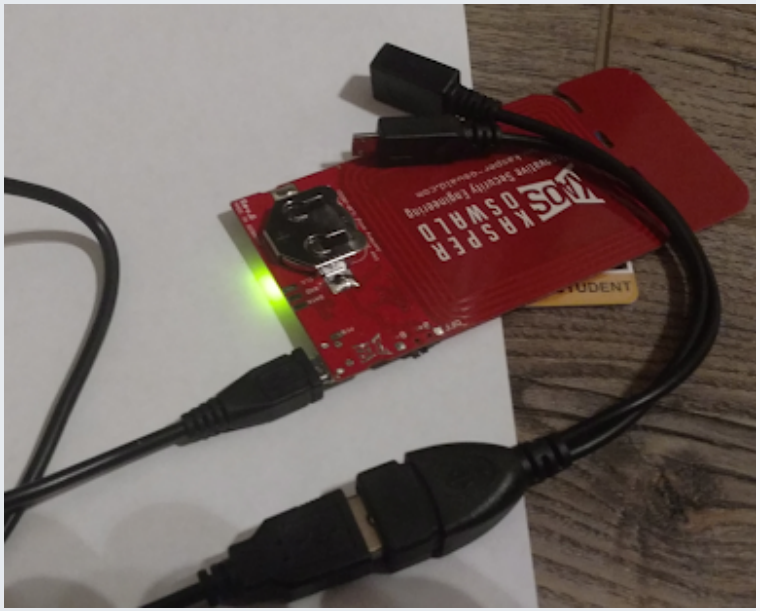
- Origins of the project
- The Chameleon Mini hardware profile and embedded firmware
- Key features of DESFire tags
- Key features of the embedded DESFire implmentation

Origins of the project

Origins of the project – Initial exploration

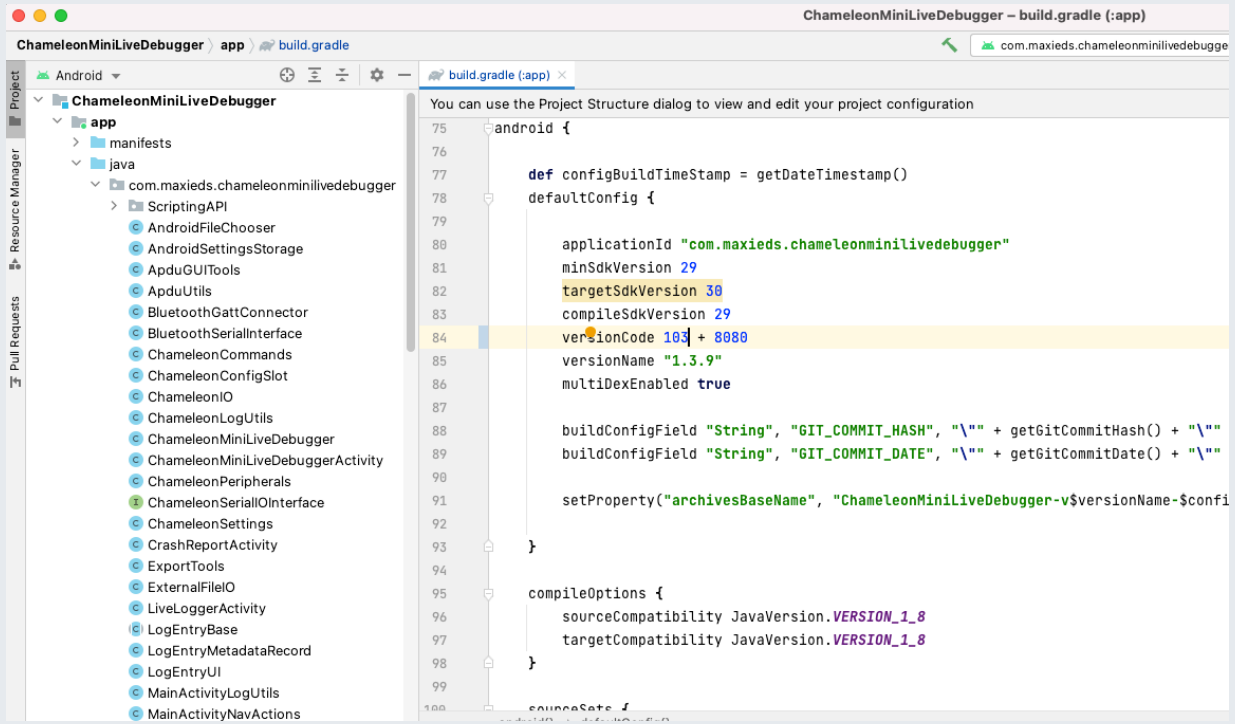


Origins of the project – Enter the Chameleon Mini




```
12291 ms < +5195 ms>:CODEC RX        (1   bytes) [26]
12292 ms <    +1 ms>:CODEC RX        (2   bytes) [9320]
12294 ms <    +2 ms>:CODEC RX        (9   bytes) [937088043c7bcbbb34]
12296 ms <    +2 ms>:CODEC RX        (2   bytes) [9520]
12298 ms <    +2 ms>:CODEC RX        (9   bytes) [95704a9849801b3abf]
12317 ms <   +19 ms>:CODEC RX        (1   bytes) [26]
12318 ms <    +1 ms>:CODEC RX        (2   bytes) [9320]
12320 ms <    +2 ms>:CODEC RX        (9   bytes) [937088043c7bcbbb34]
12322 ms <    +2 ms>:CODEC RX        (2   bytes) [9520]
12324 ms <    +2 ms>:CODEC RX        (9   bytes) [95704a9849801b3abf]
12326 ms <    +2 ms>:CODEC RX        (4   bytes) [e0803173]
12330 ms <    +4 ms>:CODEC RX        (8   bytes) [027002400100ce0c]
12344 ms <   +14 ms>:CODEC RX        (1   bytes) [26]
12346 ms <    +2 ms>:CODEC RX        (2   bytes) [9320]
12348 ms <    +2 ms>:CODEC RX        (9   bytes) [937088043c7bcbbb34]
12349 ms <    +1 ms>:CODEC RX        (2   bytes) [9520]
12351 ms <    +2 ms>:CODEC RX        (9   bytes) [95704a9849801b3abf]
12353 ms <    +2 ms>:CODEC RX        (4   bytes) [e0803173]
12358 ms <    +5 ms>:CODEC RX        (18  bytes) [0200a4040009a00000030800001000000b170]
12363 ms <    +5 ms>:CODEC RX        (16  bytes) [0300a4040c07a00000011630000011a8]
```

Origins of the project – Early prototype of the CMLD

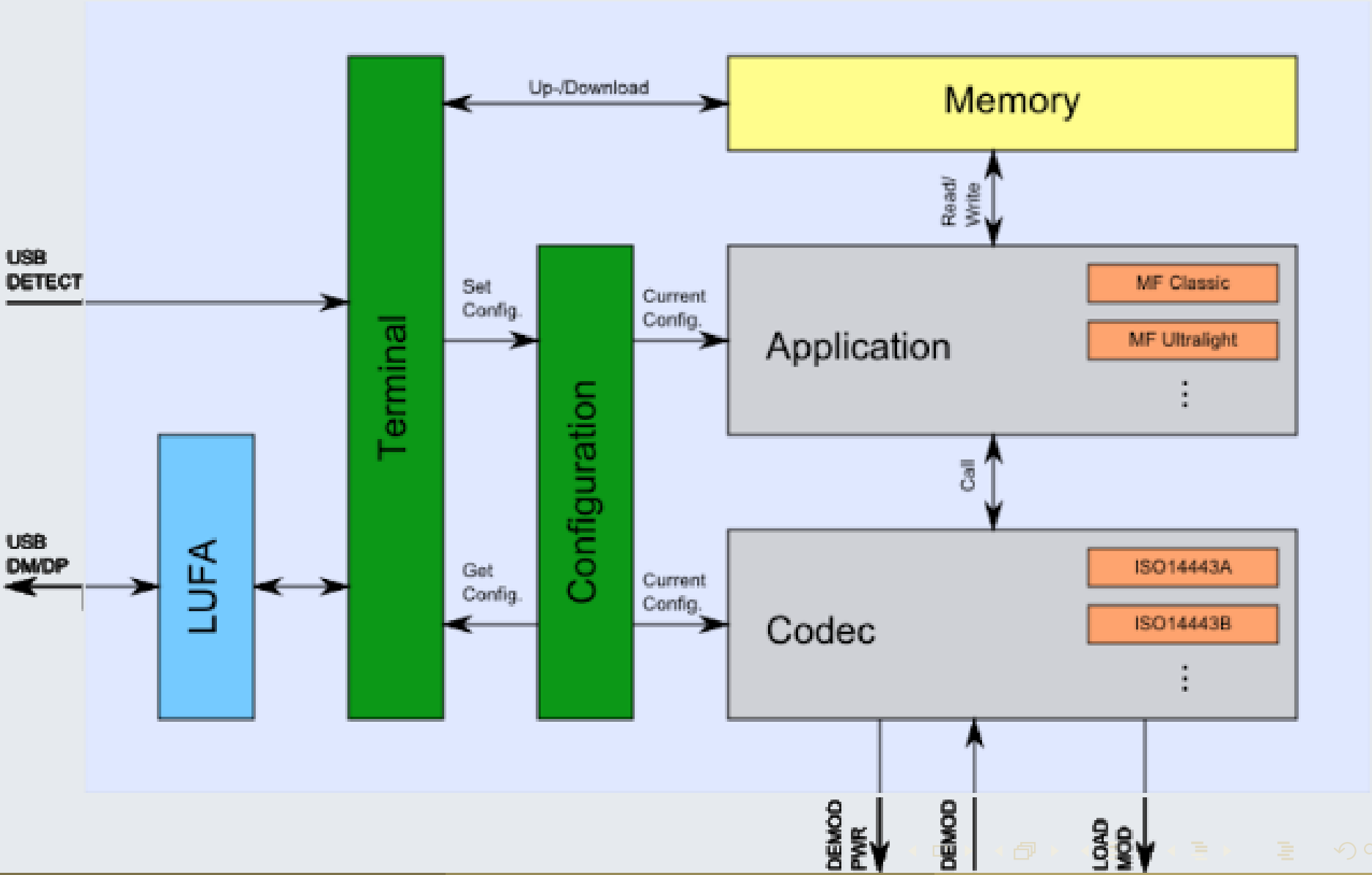


The Chameleon Mini device

The Chameleon Mini device profile – Hardware

- Modern AVR chip (`ATxmega128A4U`)
- Memory: FLASH, SRAM, EEPROM spaces and support for faster FRAM
- Embedded firmware and re-flashable bootloader (easy updates over USB)
- Memory mapping of the integrated RF hardware for easy access from C/ASM firmware sources
- Accelerated hardware support for AES and DES cryptographic engines
- Serial data transfer over wired micro-USB

Chameleon Mini Firmware



DESFire tags

Key Features

- Mutiple generations of DESFire tags
- Larger memory sizes than most NFC tags
- Use of modern cryptographic algorithms for secure data exchange
- Data messages optionally padded with crytographically hashed bytes for data integrity
- Even though DESFire tags are popular in applications, they are hard to interface with due to proprietary specs

Filesystem: Organization and internal storage types

- Files grouped by allocations of the physical memory into top-level subdirectories called applications (indexed by 3-byte AID)
- Support for several file types
- Access permissions on the files provide more security for secret binary key data

Commands and native instruction support

- APDU messages are used to exchange data (native, wrapped, ISO standard wrapped formats)
- Think of APDU messages as an "assembly language" for NFC exchanges

---

PCD-to-PICC wrapped APDU data exchange format:

CLA	INS	P_1	P_2	L_c	Data Bytes	L_e
0x90	command code	0x00	0x00	variable length of data	command data	0x00

PICC-to-PCD format:

Data Bytes	SW1	SW2 (Status)
DESFire command response data	0x91	0xYY

Supported command codes – Some examples

Command Long Name	INS	Description
AUTHENTICATE	0x0A	Legacy mode authentication
AUTHENTICATE_ISO	0x1A	ISO authentication with 3DES
AUTHENTICATE_AES	0xAA	Standard AES authentication
CHANGE_KEY_SETTINGS	0x54	Modify PICC master key properties
SET_CONFIGURATION	0x5C	Used to configure DESFire card or application specific attributes
CHANGE_KEY	0xC4	Changes the key data stored on the PICC
GET_KEY_VERSION	0x64	Returns the active key version stored on the PICC
CREATE_APPLICATION	0xCA	Creates new applications by unique AID
DELETE_APPLICATION	0xDA	Non-restorable deletion operation
GET_APPLICATION_IDS	0x6A	Returns a list of all AID codes stored on the PICC
FREE_MEMORY	0x6E	Returns the total free memory on the tag in bytes
GET_DF_NAMES	0x6D	Obtain the ISO7816-4 DF names associated with the tag
GET_KEY_SETTINGS	0x45	Get permissions data and format for PICC and application master keys
SELECT_APPLICATION	0x5A	Select a specific application by AID for further access

A more complete listing of supported commands is found in the source code; alternately, the data sheets linked in the bibliography archived by incidentally lucky NFC researchers give command syntax and argument details precisely.

Data exchanges with the Chameleon DESFire configuration

```
>>> Select Application By AID:
    -> 90 5a 00 00 03 00 00 00 | 00
    <- 91 00
>>> Start AES Authenticate:
    -> 90 aa 00 00 01 00 00
    <- 54 b8 9e fe 19 9b c6 a5 | fd 8f 00 be c1 23 99 c0 | 91 af
    -> 90 af 00 00 10 df a0 79 | 13 59 ac 4c 75 5f 81 69 |
       bc 9c 3e c6 7e 00
    <- a9 e2 79 42 11 63 9c 14 | 07 b3 02 2f 2e 4b 2e c5 | 91 00
>>> Get AID List From Device:
    -> 90 6a 00 00 00 00
    <- 77 88 99 01 00 34 91 00
>>> CreateApplication command:
    -> 90 ca 00 00 05 77 88 99 | 0f 03 00
    <- 91 de
>>> Get AID List From Device:
    -> 90 6a 00 00 00 00
    <- 77 88 99 01 00 34 91 00
```

More complete examples of data exchanges using these commands are found in the `LibNFC` *testing code within the Chameleon mini main firmware on the GitHub/emsec/ChameleonMini repository (in the* `Software/DESFireLibNFCTesting` *directory).*

An Embedded Open Source DESFire Stack for the Chameleon Mini

Extensions of the firmware sources to support DESFire tags

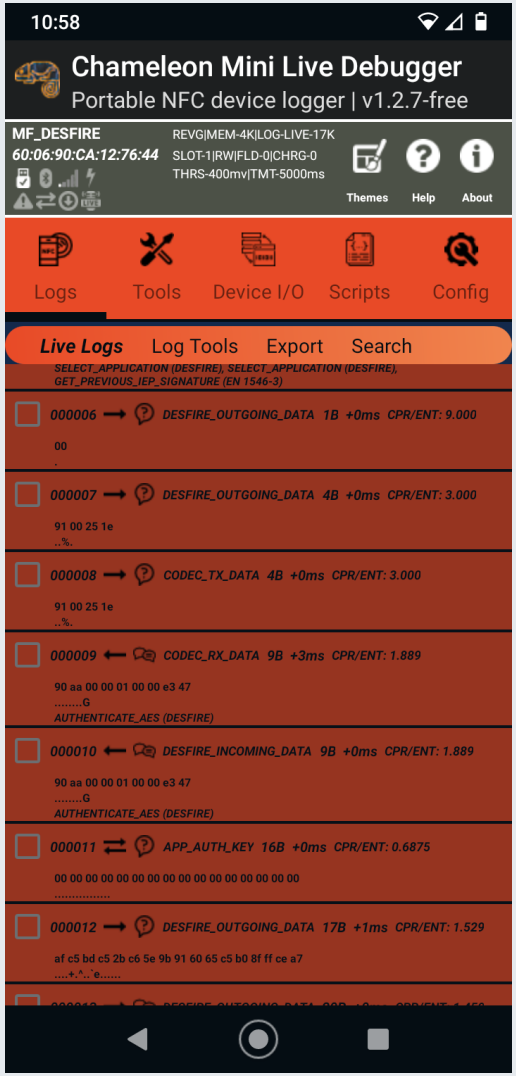
- New native AES support using hardware acceleration
- Extensions of DES and 3DES support to the firmware
- Small changes to the codec layer of the firmware
- Other enhancements and bug fixes
- Support for extended Chameleon terminal commands for DESFire emulation modes

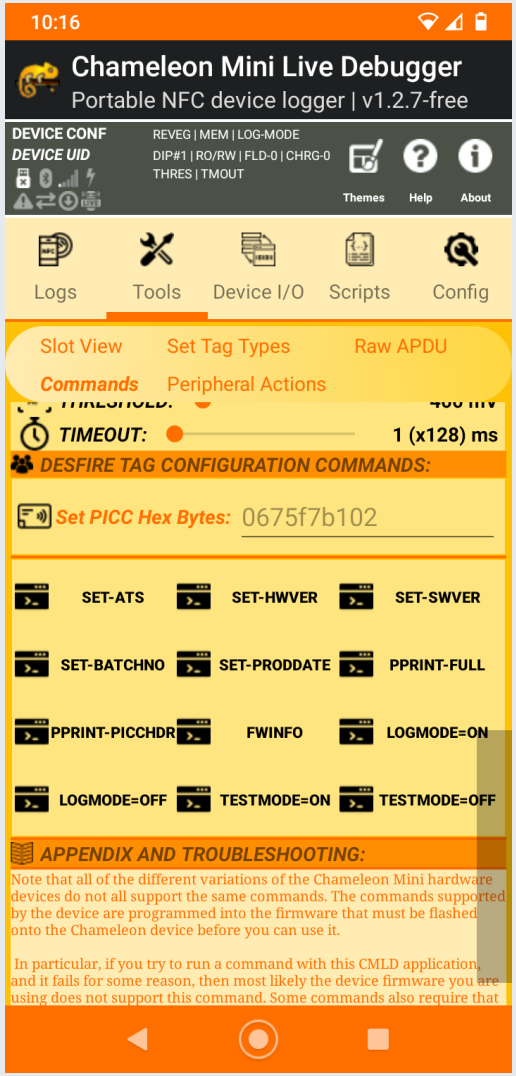
Terminal configuration of DESFire emulation support

```
> CONFIG=MF_DESFIRE
> DF_SETHDR=ATS 0675F7B102
> UID=2377000B99BF98

--------------------------------

DF_SETHDR=ATS xxxxxxxxxx
DF_SETHDR=HardwareVersion xxxx
DF_SETHDR=SoftwareVersion xxxx
DF_SETHDR=BatchNumber xxxxxxxxxx
DF_SETHDR=ProductionDate xxxx
```





DESFire emulation support – Anti-collision loop

```
NFC reader: SCM Micro / SCL3711-NFC&RW opened

Sent bits:     26 (7 bits)
Received bits: 03  44
Sent bits:     93  20
Received bits: 88  23  77  00  dc
Sent bits:     93  70  88  23  77  00  dc  4b  b3
Received bits: 04
Sent bits:     95  20
Received bits: 0b  99  bf  98  b5
Sent bits:     95  70  0b  99  bf  98  b5  2f  24
Received bits: 20
Sent bits:     e0  50  bc  a5
Received bits: 75  77  81  02  80
Sent bits:     50  00  57  cd

Found tag with
UID: 2377000b99bf98
ATQA: 4403
SAK: 20
ATS: 75  77  81  02  80
```

Terminal screenshot of the DESFire source code

MifareDESFire.c (~/Desktop/GATechCOVIDReliefProjectCode/ChameleonMini/Firmware/Chameleon-Mini/Ap

```
t16_t MifareDesfireAppProcess(uint8_t *Buffer, uint16_t BitCount) {
 uint16_t ByteCount = (BitCount + BITS_PER_BYTE - 1) / BITS_PER_BYTE;
 uint16_t ReturnedBytes = 0;
 LogEntry(LOG_INFO_DESFIRE_INCOMING_DATA, Buffer, ByteCount);
 if (ByteCount >= 8 && DesfireCLA(Buffer[0]) && Buffer[2] == 0x00 &&
         Buffer[3] == 0x00 && Buffer[4] == ByteCount - 8) {
     DesfireCmdCLA = Buffer[0];
     uint16_t IncomingByteCount = (BitCount + BITS_PER_BYTE - 1) / BITS_PER_BYTE;
     uint16_t UnwrappedBitCount = DesfirePreprocessAPDU(ActiveCommMode, Buffer, IncomingByteCount) * BITS_PE
     uint16_t ProcessedBitCount = MifareDesfireProcess(Buffer, UnwrappedBitCount);
     uint16_t ProcessedByteCount = (ProcessedBitCount + BITS_PER_BYTE - 1) / BITS_PER_BYTE;
     ProcessedByteCount = DesfirePostprocessAPDU(ActiveCommMode, Buffer, ProcessedByteCount);
     LogEntry(LOG_INFO_DESFIRE_OUTGOING_DATA, Buffer, ProcessedByteCount);
     return ISO14443AStoreLastDataFrameAndReturn(Buffer, ProcessedByteCount * BITS_PER_BYTE);
 }
 Iso7816CmdType = IsWrappedISO7816CommandType(Buffer, ByteCount);
 if (Iso7816CmdType != ISO7816_WRAPPED_CMD_TYPE_NONE) {
     DesfireCmdCLA = (Iso7816CmdType == ISO7816_WRAPPED_CMD_TYPE_STANDARD) ? Buffer[2] : DESFIRE_NATIVE_CLA;
     uint8_t ISO7816PrologueBytes[2];
     memcpy(&ISO7816PrologueBytes[0], Buffer, 2);
     if (Iso7816CmdType == ISO7816_WRAPPED_CMD_TYPE_STANDARD) {
         memmove(&Buffer[0], &Buffer[2], ByteCount - 2);
         ByteCount = ByteCount - 2;
     } else if (Iso7816CmdType == ISO7816_WRAPPED_CMD_TYPE_PM3_ADDITIONAL_FRAME) {
         Buffer[0] = DesfireCmdCLA;
         Buffer[1] = STATUS_ADDITIONAL_FRAME;
         if (ByteCount > 3) {
             memmove(&Buffer[5], &Buffer[3], ByteCount - 3);
         }
         Buffer[2] = 0x00;
         Buffer[3] = 0x00;
         Buffer[4] = ByteCount - 5;
         ByteCount += 2;
     } else if (Iso7816CmdType == ISO7816_WRAPPED_CMD_TYPE_PM3RAW) {
         /* Something like the following (for PM3 raw ISO auth):
          * 0a 00 1a 00 CRC1 CRC2 -- first two are prologue -- last two are checksum
          */
         Buffer[0] = DesfireCmdCLA;
         Buffer[1] = Buffer[2];
```

ISO authentication with the PM3 (Support added in 2022)

```
[=] Waiting for Proxmark3 to appear...
[=] Session log /Users/mschmidt34/.proxmark3/logs/log_20220413.txt
[+] loaded from JSON file /Users/mschmidt34/.proxmark3/preferences.json
[=] Using UART port /dev/tty.usbmodemiceman1
[=] Communicating with PM3 over USB-CDC


  8888888b.  888b     d888  .d8888b.
  888   Y88b 8888b   d8888 d88P  Y88b
  888    888 88888b.d88888      .d88P
  888   d88P 888Y88888P888     8888"
  8888888P"  888 Y888P 888      "Y8b.
  888        888  Y8P  888 888    888
  888        888   "   888 Y88b  d88P
  888        888       888  "Y8888P"    [ ❄ ]


  [ Proxmark3 RFID instrument ]

    MCU....... AT91SAM7S512 Rev B
    Memory.... 512 Kb ( 53% used )

    Client.... Iceman/master/v4.14831-511-g78e99d3e3 2022-03-30 09:51:46
    Bootrom... Iceman/master/v4.14831-404-gab5213126-dirty-unclean 2022-03-28 16:40:45
    OS........ Iceman/master/v4.14831-404-gab5213126-dirty-unclean 2022-03-28 16:41:00
    Target.... PM3 GENERIC


[!] ⚠  --> ARM firmware does not match the source at the time the client was compiled
[!] ⚠  --> Make sure to flash a correct and up-to-date version

[[usb] pm3 --> hf mfdes auth -n 0 -t 3tdea -k 000000000000000000000000000000000000000000000000 -v -c native -a
[=] Key num: 0 Key algo: 3tdea Key[24]: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[=] Secure channel: n/a Command set: native Communication mode: plain
[+] Setting ISODEP -> inactive
[+] Setting ISODEP -> NFC-A
[=] AID 000000 is selected
[=] Auth: cmd: 0x1a keynum: 0x00
[+] raw>> 1A 00
[+] raw<< AF EE 91 30 1E E8 F5 84 D6 C7 85 1D 05 65 13 90 A6 C6 D5
[#] encRndB: EE 91 30 1E E8 F5 84 D6
[#] RndB: CA FE BA BE 00 11 22 33
[#] rotRndB: FE BA BE 00 11 22 33 CA FE BA BE 00 11 22 33 CA
[#] Both    : 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 FE BA BE 00 11 22 33 CA FE BA BE 00 11 22 33 CA
[+] raw>> AF 30 EB 55 F3 29 39 04 96 77 88 CE EF 33 A3 C8 7B 18 66 1A F1 62 78 A0 28 53 84 67 98 7C BB DB 03
[+] raw<< 00 9B 71 57 8F FB DF 80 A8 F6 EF 33 4A C6 CD F9 7A 7D BE
[=] Session key : 01 02 03 04 CA FE BA BE 07 08 09 10 22 33 CA FE 13 14 15 16 00 11 22 33
[=] Desfire  authenticated
[+] PICC selected and authenticated succesfully
[+] Context:
[=] Key num: 0 Key algo: 3tdea Key[24]: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[=] Secure channel: ev1 Command set: native Communication mode: plain
[=] Session key [24]: 01 02 03 04 CA FE BA BE 07 08 09 10 22 33 CA FE 13 14 15 16 00 11 22 33
[=]     IV [8]: 00 00 00 00 00 00 00 00
[+] Setting ISODEP -> inactive
```

Challenges with the implementation during development

- Six to eight months of developmentto complete the inital stages of the project
- Considerations to optimize and organize handling of memory constraints in the embedded AVR chip
- Hardware acceleration needed for AES/3DES crypto routines: Existing crypto libraries for AVR not fast enough
- Semi-complicated, quasi-linked pointer-based structure organization to efficiently store DESFire filesystem and metadata

Concluding Remarks

Summary and accomplishments

- Compatible with most external USB NFC readers, LibNFC and PM3 devices
- DESFire support on the Chameleon Mini is by far the most requested single feature from users
  - **GitHub/emsec/ChameleonMini watchers:** 129
  - **GitHub/emsec/ChameleonMini stars:** 1344
  - **GitHub/emsec/ChameleonMini forks:** 342
- CMLD on Google Play Store peaked at ≈ 500 (free) and ≈ 50 (paid) users internationally
  - **GitHub/maxieds/ChameleonMiniLiveDebugger stars:** 76
  - **GitHub/maxieds/ChameleonMiniLiveDebugger forks:** 15

Funding sources and support for the project

- Initial sources for the DESFire Chameleon firmware are due to Dmitry Janushkevich (from 2017)
- Professor Josephine Yu in the School of Math at GA Tech in the US
- Georgia Tech for supporting me as a RA in the Spring of 2022 through the university's COVID-19 relief funding
- The original Kasper and Oswald (KAOS) developers of the Chameleon Mini hardware and software
- David Oswald from the University of Birmingham in the UK